OpenSatKit Enables Quick Startup for CubeSat Missions

David McComas, NASA Goddard Space Flight Center

Ryan Melton, Ball Aerospace Corporation

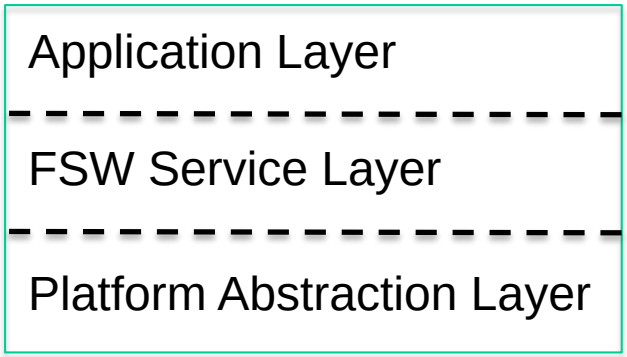
August 7, 2017

CubeSat Flight Software Challenges

- Flight software complexity does not scale down with size
- Spacecraft budgets do scale down with size
- Flight software must be reliable for mission success
- People, Processors, and Projects change

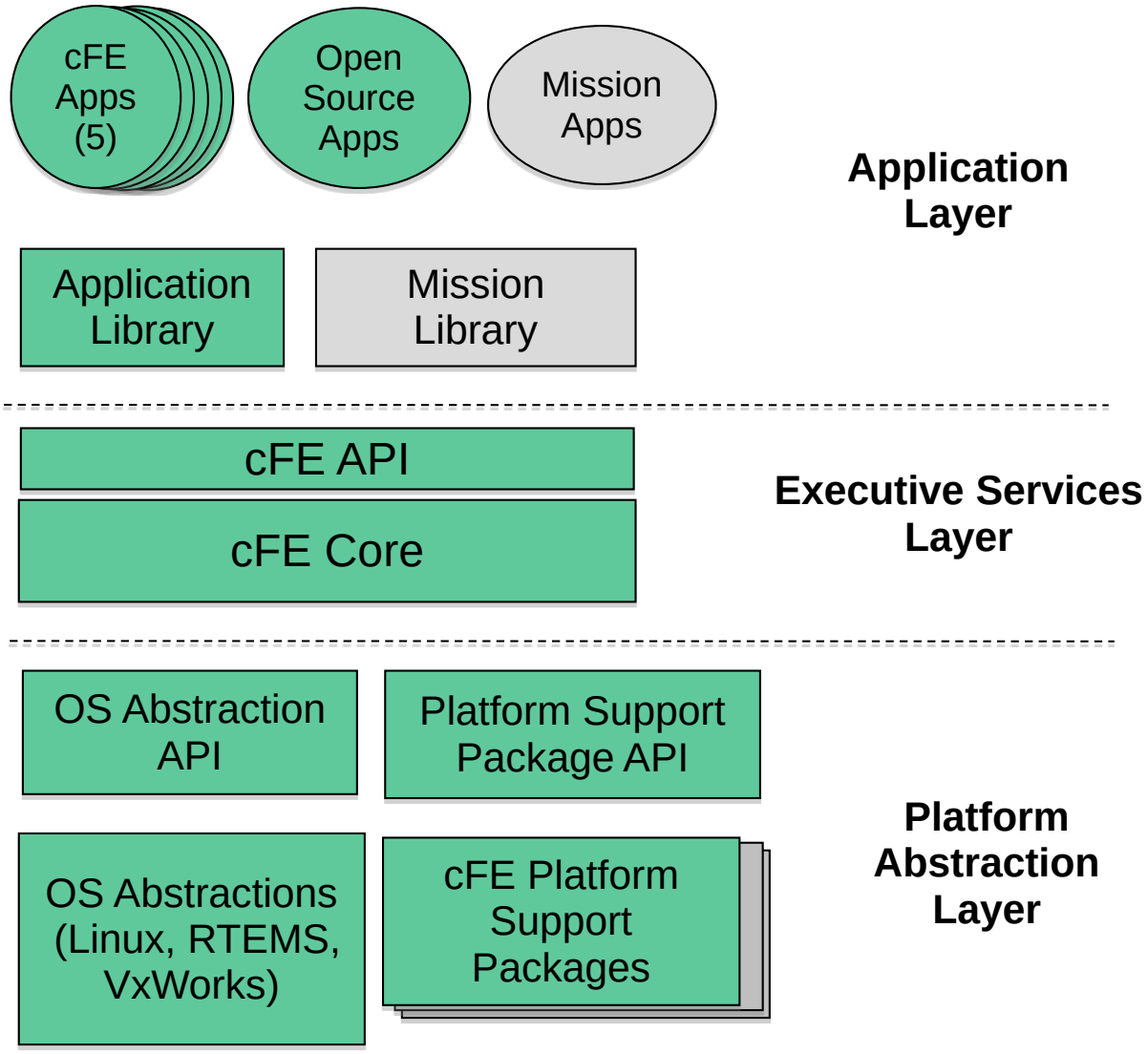
Core Flight System as a Solution

- The cFS is a re-usable spacecraft flight software architecture that provides a portable and extendable platform with a product line deployment model
    - Platform Abstraction Layer supports portability
    - Applications provide mission functionality
    - Compile-time configuration parameters and run-time command/table parameters flexibility and scalability

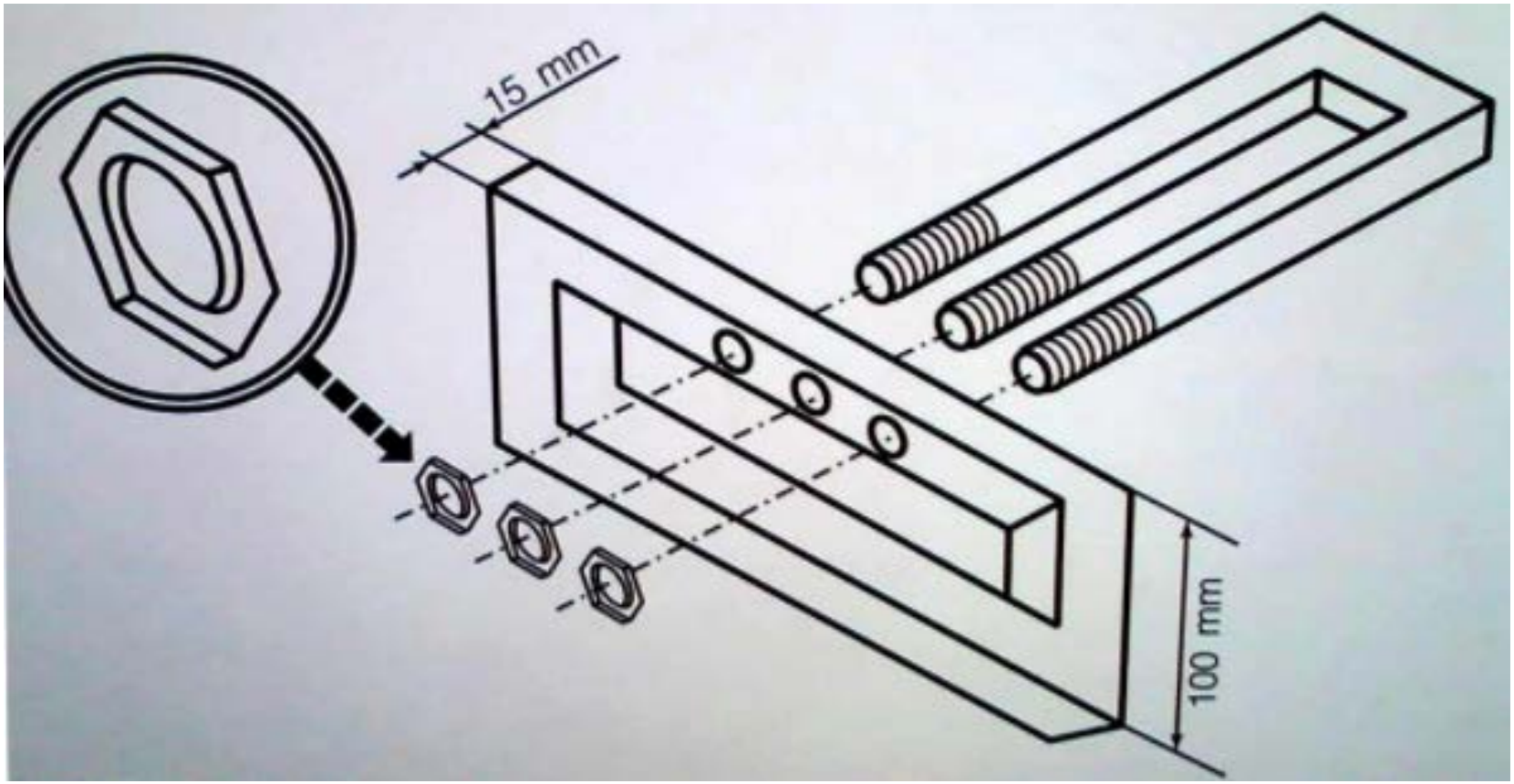


- NASA Goddard released cFS as open source in 2015
- Changes controlled by a NASA multi-center configuration control board

cFS Layer Architecture



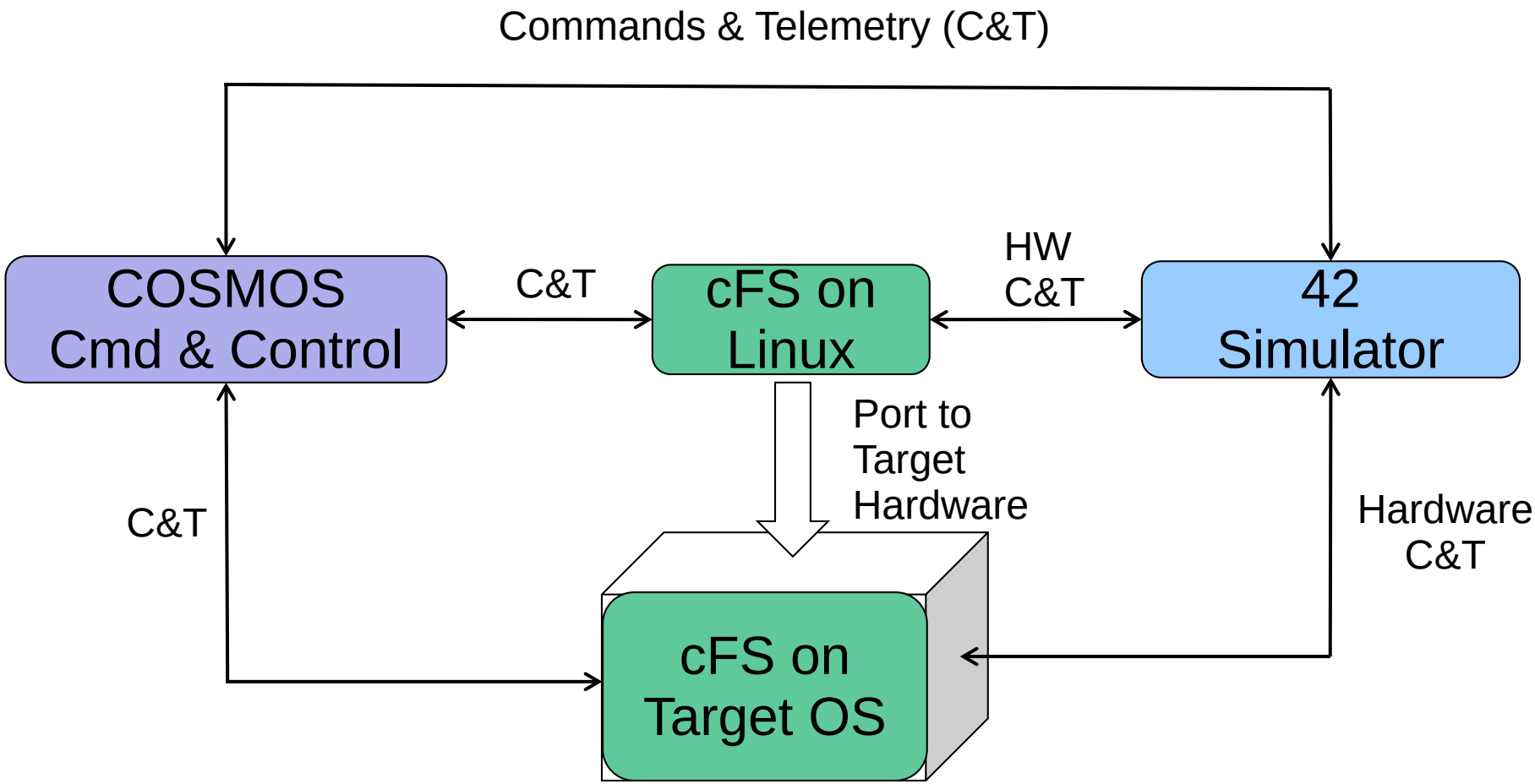
Some Assembly Required



- Okay, I down loaded the Core Flight Executive, now what?
- What apps do I need?
- How do I send commands and receive telemetry?
- How should I configure the system?
- How do I write a new app?

OpenSatKit as the Solution

- Provides a turnkey solution for COSMOS-cFS



COSMOS Introduction

WHAT IS BALL AEROSPACE COSMOS?

An open source command and control system that lets you send commands, visualize data (graphs, textual display), write automated scripts, analyze log files, monitor telemetry, and more

Ball Aerospace

COSMOS

COSMOS For OpenSatKit

HOW DOES COSMOS APPLY TO OPENSATKIT?

1. COSMOS comes preconfigured and ready to communicate with the NASA CFS
2. COSMOS TlmViewer setup as a custom interface to the flight software
3. COSMOS Logs all Commands and Telemetry
4. All of the 15+ COSMOS Tools are ready to use with the flight software to write test/operations procedures, graph telemetry, edit binary tables, analyze logged data and more

COSMOS News For SmallSat

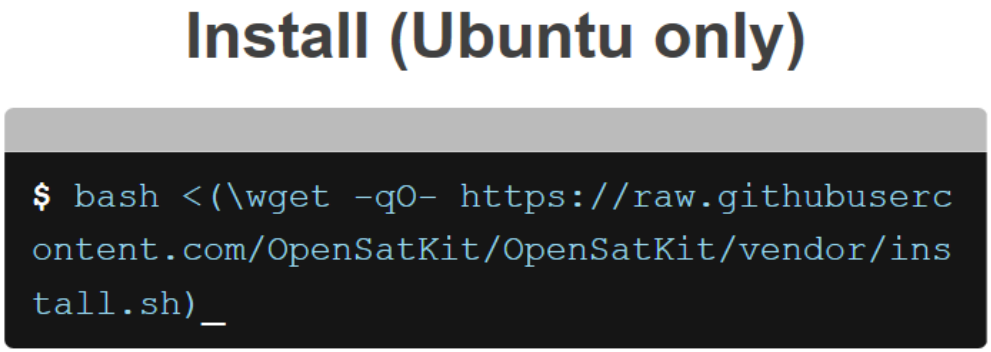
WHAT'S NEW WITH COSMOS?

1. COSMOS 4.0 just released with over 45 change requests Included
2. Two New Tools
    1. Config Editor
    2. Command Sequence
3. New Protocol System, View Raw for Interface, Performance Improvements, and a lot more
4. OpenSatKit support coming in a few weeks

Ball Aerospace
COSMOS

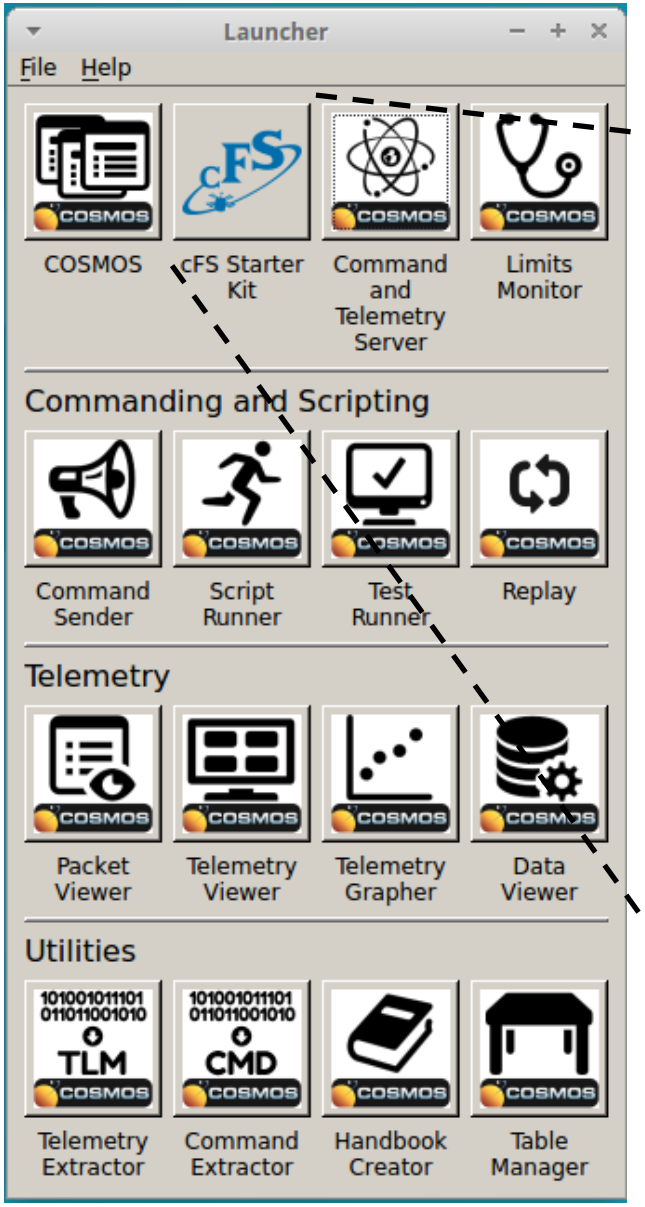
Installing OpenSatKit

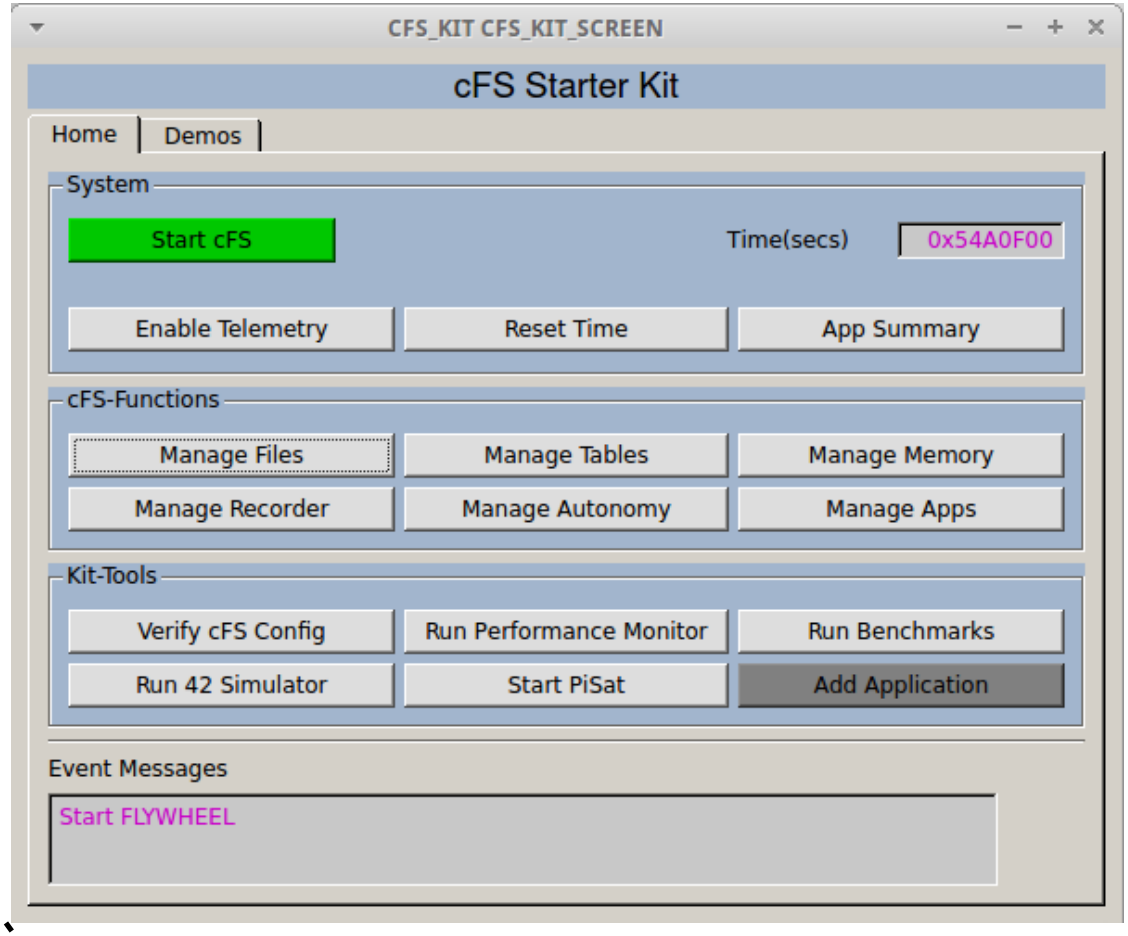
1. Create an Ubuntu host environment
2. Goto https://opensatkit.github.io



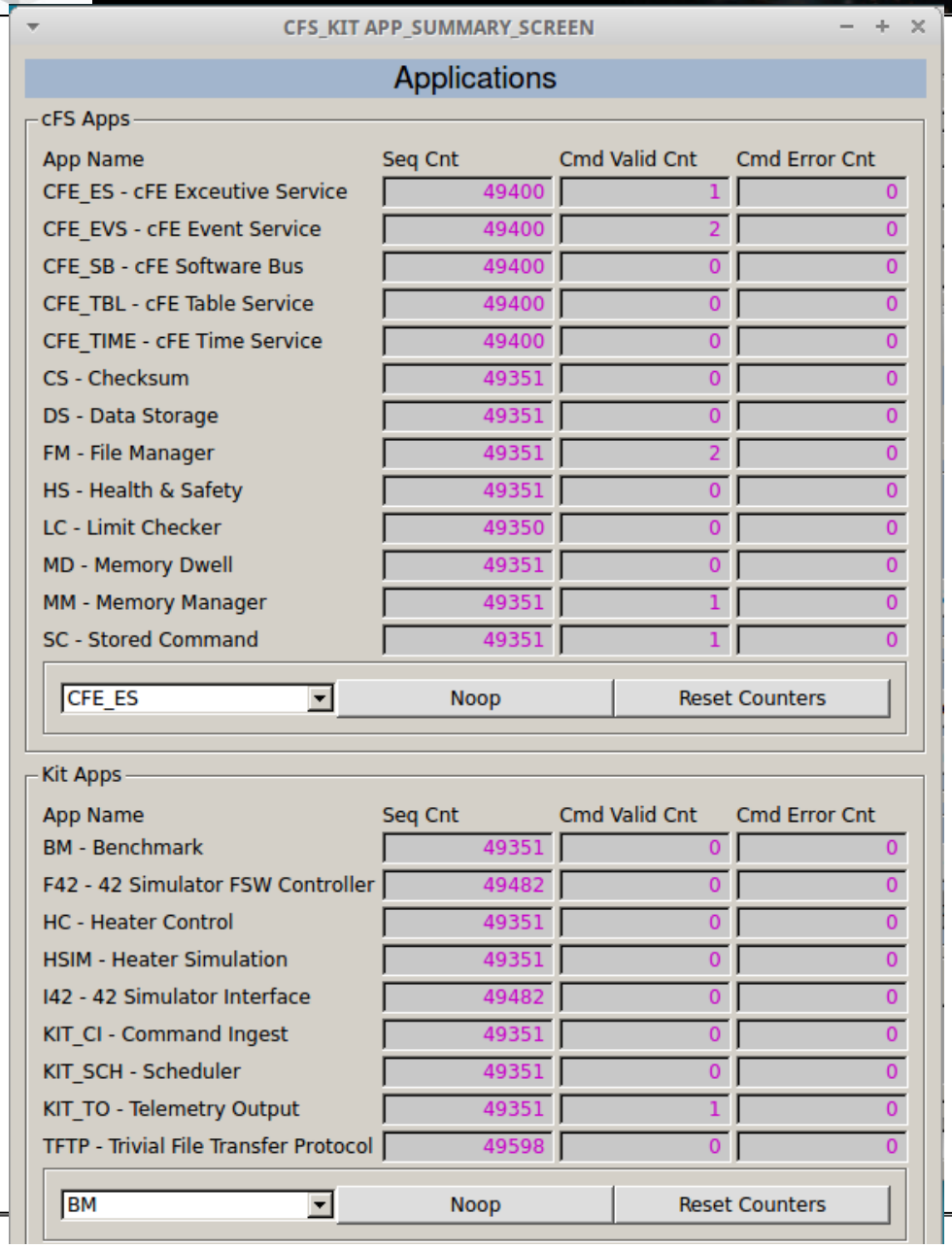
3. Run the installation script that automatically launches COSMOS
4. Start the Core Flight System from the main screen

Starting the Kit



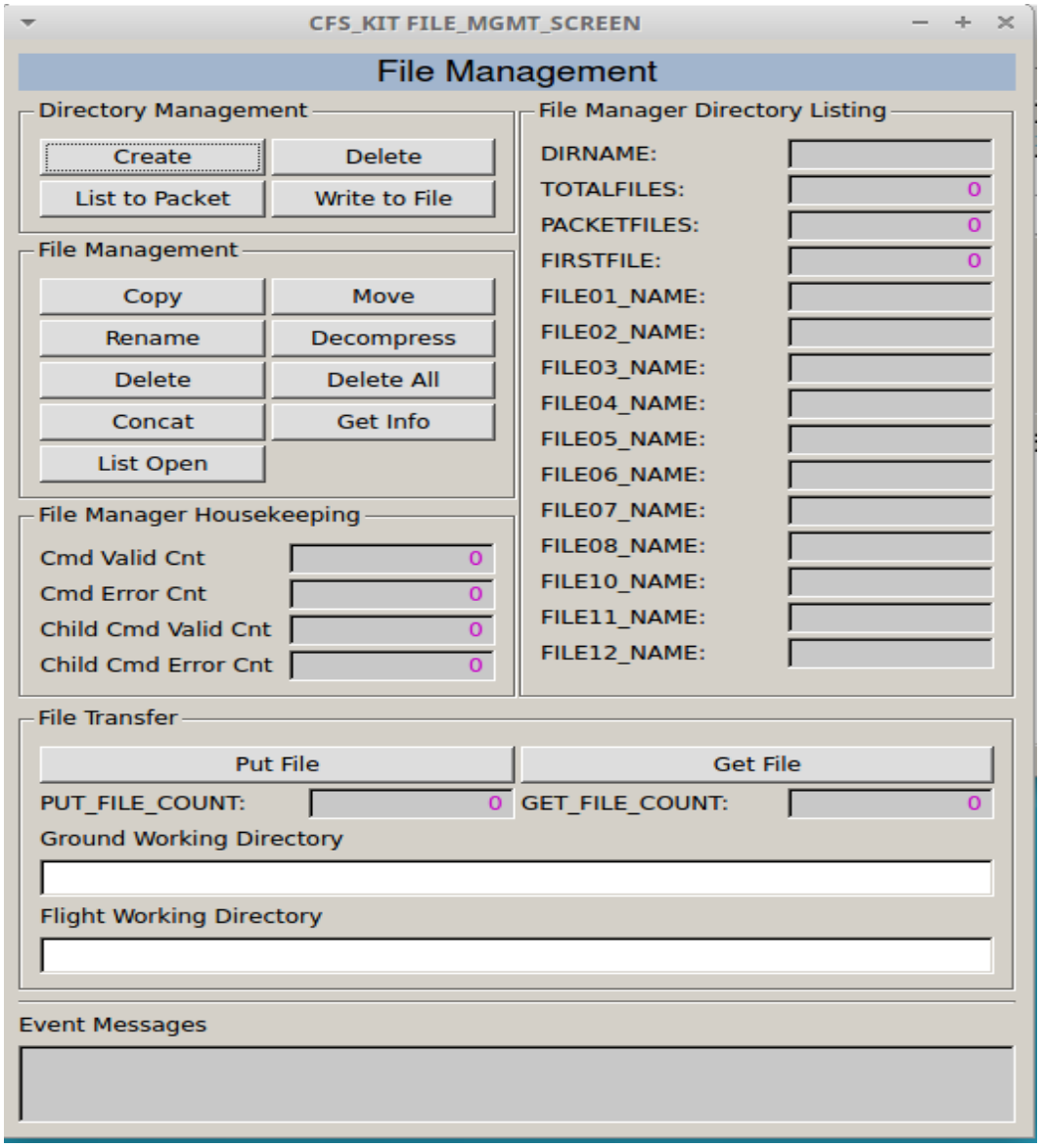


Preconfigured Applications



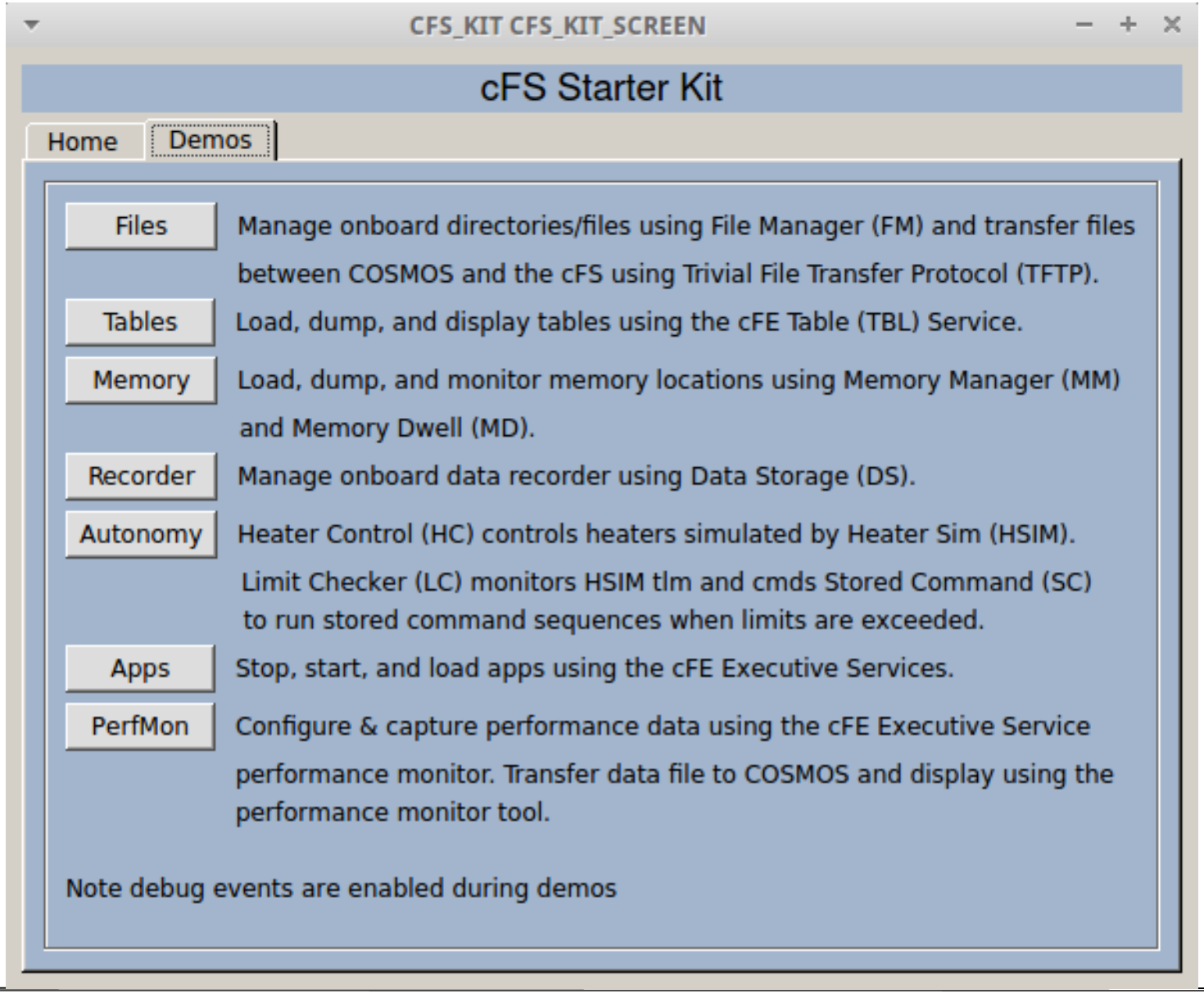
- Complete Cmd & Tlm Definitions
- NASA Goddard Dellingr CubeSat uses 5 cFE & 6 cFS apps
- Kit apps
  - Provide desktop functionality
  - Not intended for flight

Functional Organization



- File Management
  - File Transport App
  - File Management App

Built In Demos



PiSat Target

Next Steps

- Feature enhancements
  - Enhanced table tool support for current cFS apps
  - Add CCSDS File Delivery Protocol (CFDP) app
  - Create application plug in model
  - Create platform target integration model
  - Enhanced simulator support
- Collaborate

Conclusion

- NASA Booths 84 & 85
- Ball Aerospace Booths 65 & 66
- Classroom kit demos Tues & Wed
- **https://opensatkit.github.io**

Special Thanks

- NASA Core Flight System Configuration Control Board
- Jason Thomas of Ball Aerospace for COSMOS work
- Eric Stoneking of NASA Goddard for 42 Simulator
- Alan Cudmore PiSat
- University of Maryland Summer Interns
  - Antonio Heard
  - Johann Miller
  - Will Yeager

Backup Slides

State of the Community Communication

- Mailing Lists
  - **cfs-community@lists.nasa.gov**
    - Contains all members
  - **cfs-community-ccb@lists.nasa.gov**
    - CCB members

- Public Websites
  - **https://cfs.gsfc.nasa.gov/**
    - General information and links to all open source code and documents on Sourceforge
  - **https://sourceforge.net/projects/xxx**
    - Multiple projects for different cFS components

- Restricted access (requires NDC account)
  - **https://nsckn.nasa.gov/Community**
    - NESC hosted server containing discussion forums, documents, meeting notes…
    - Approved for ITAR and Sensitive But Unclassified (SBU) material
  - **https://babelfish.arc.nasa.gov/**
    - ARC hosted server used for inter-center collaboration
    - Git andTrac used for source code configuration management and change requests
    - Not approved for ITAR material

Questions? Contact:

- CFS Program Manager
  - David McComas - NASA GSFC/Code 580 Software Division
    david.c.mccomas@nasa.gov
    301.286.9038
- CFS Lead Architect
  - Jonathan Wilmot - NASA GSFC/Code 582 Flight Software Branch
    jonathan.j.wilmot@nasa.gov
    301.286.2623
- CFS Product Development Lead
  - Susie Strege - NASA GSFC/Code 582 Flight Software Branch
    susie.strege@nasa.gov
    301.286.3829
- OSAL Product Development Lead
  - Alan Cudmore - NASA GSFC/Code 582 Flight Software Branch
    alan.p.cudmore@nasa.gov
    301.286.5809

OS Abstraction Layer Platforms

Operating System	OSAL Version	Status	Target
POSIX/Linux	4.1.1	Production	Desktop Dev. use CentOS 6.x/Ubuntu 14.04 32 bit
RTEMS	4.1.1	Production	Flying on MMS Mission RTEMS 4.10/Coldfire
VxWorks	4.1.1	Production	Flying on GPM Mission vxWorks 6.4/PowerPC
FreeRTOS	4.2.x	In Dev.	GSFC Dellingr CubeSat Mission FreeRTOS/Arm
VxWorks 6.x SMP	4.3.x	In Dev.	vxWorks 6.7 LEON3 Dual Core
ARINC653	4.3.x	In Dev.	Green Hills Integrity OS
RTEMS 4.12+SMP	Future	Future	Future Release
Xenomai Linux	Future	Future	Future Release

cFE 6.5 Platform Support Packages

Board/Platform	OSAL Operating System	Status
CentOS/Ubuntu Linux Desktop	POSIX/Linux	Used on a balloon mission Common initial development/test environment
MMS Custom C&DH Coldfire	RTEMS	1 year in flight on MMS Mission
GPM RAD750	VxWorks	2 years in flight on GPM Mission
Gomspace Nanomind ARM CubeSat	FreeRTOS	Under development for GSFC Dellingr CubeSat Mission
GSFC MUSTANG Dual Core LEON3	VxWorks SMP	Under development for GSFC MUSTANG Dual Core LEON3 architecture

Example Mission Code Metrics
Global Precipitation Measurement (GPM)

- Noteworthy items
  - \+ cFE was very reliable and stable
  - \+ Easy rapid prototyping with heritage code that was cFE compliant
  - \+ Layered architecture has allowed COTS lab to be maintained through all builds
  - \- Addition of PSP changed build infrastructure midstream
- Lines of Code Percentages:

Source	Percentage
BAE	0.3
EEFS	1.7
OSAL	2.1
PSP	1.0
cFE	12.4
GNC Library	1.6
CFS Applications	23.5
Heritage Clone & Own	38.9
New Source	18.5

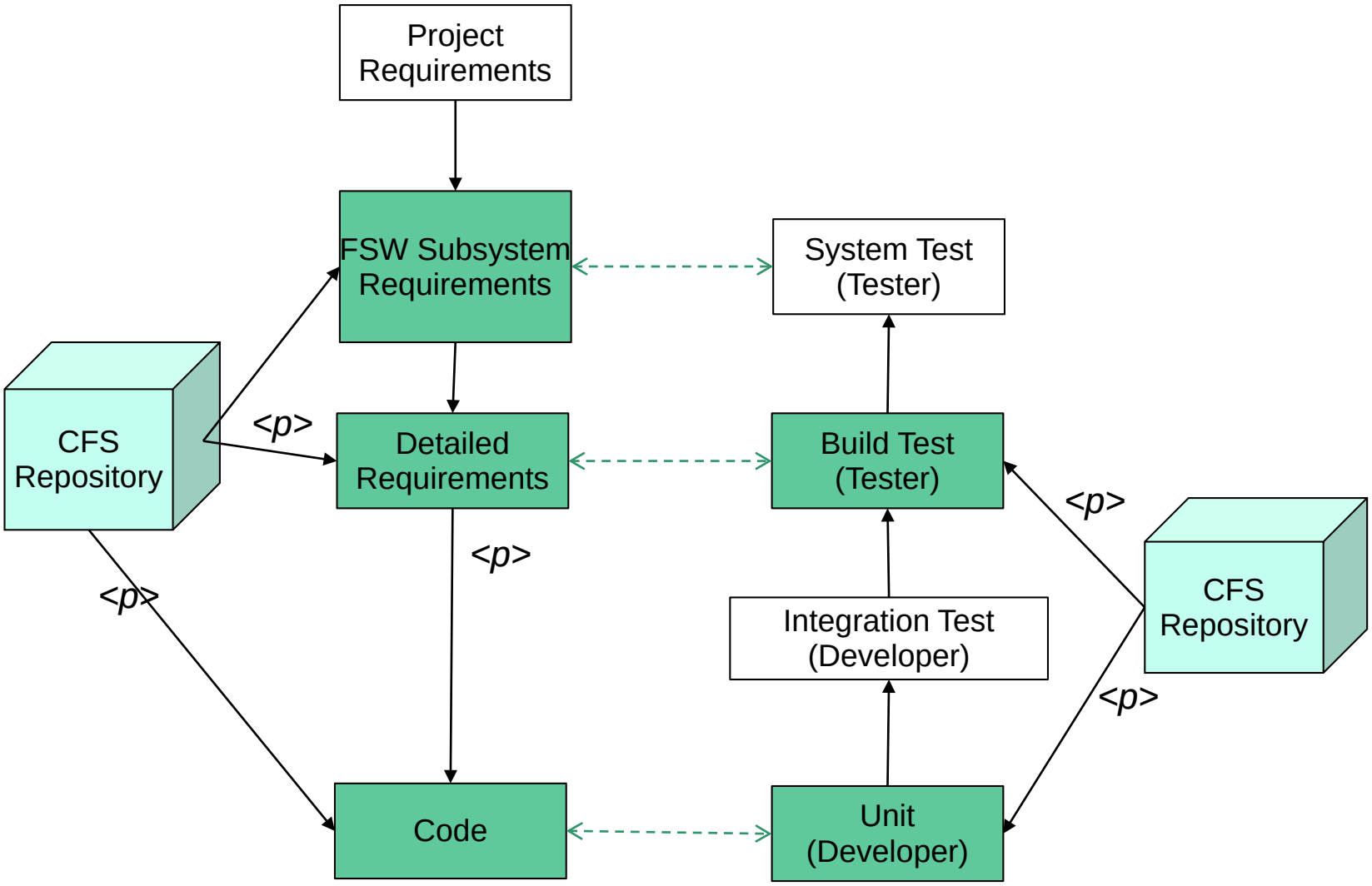
cFS Metrics

cFE/ App	Logical Lines of Code (non-table)	Config. Parameters	EEPROM (bytes)
cFE	12,930	General: 17 Executive Service: 46 Event Service: 5 Software Bus: 29 Table Service: 10 Time Service: 32	341,561
CFDP	8,559	33	85,812
Checksum	2,873	15	35,242
Data Storage	2,429	27	40,523
File Manager	1,853	22	16,272
Health & Safety	1,531	45	15071
House-Keeping	575	8	8.059
Limit Checker	2,074	13	31,026
Memory Dwell	1,035	8	8,617
Memory Manager	1,958	25	15,840
Scheduler	1,164	19	35,809
Stored Command (124 command sequences)	2,314	26	104,960

cFS Applications

Application	Function
CFDP	Transfers/receives file data to/from the ground
Checksum	Performs data integrity checking of memory, tables and files
Command Ingest Lab	Accepts CCSDS telecommand packets over a UDP/IP port
Data Storage	Records housekeeping, engineering and science data onboard for downlink
File Manager	Interfaces to the ground for managing files
Housekeeping	Collects and re-packages telemetry from other applications.
Health and Safety	Ensures that critical tasks check-in, services watchdog, detects CPU hogging, and calculates CPU utilization
Limit Checker	Provides the capability to monitor values and take action when exceed threshold
Memory Dwell	Allows ground to telemeter the contents of memory locations. Useful for debugging
Memory Manager	Provides the ability to load and dump memory.
Software Bus Network	Passes Software Bus messages over Ethernet
Scheduler	Schedules onboard activities via (e.g. HK requests)
Scheduler Lab	Simple activity scheduler with a one second resolution
Stored Command	Onboard Commands Sequencer (absolute and relative).
Telemetry Output Lab	Sends CCSDS telemetry packets over a UDP/IP port

Lifecycle Artifact Reuse



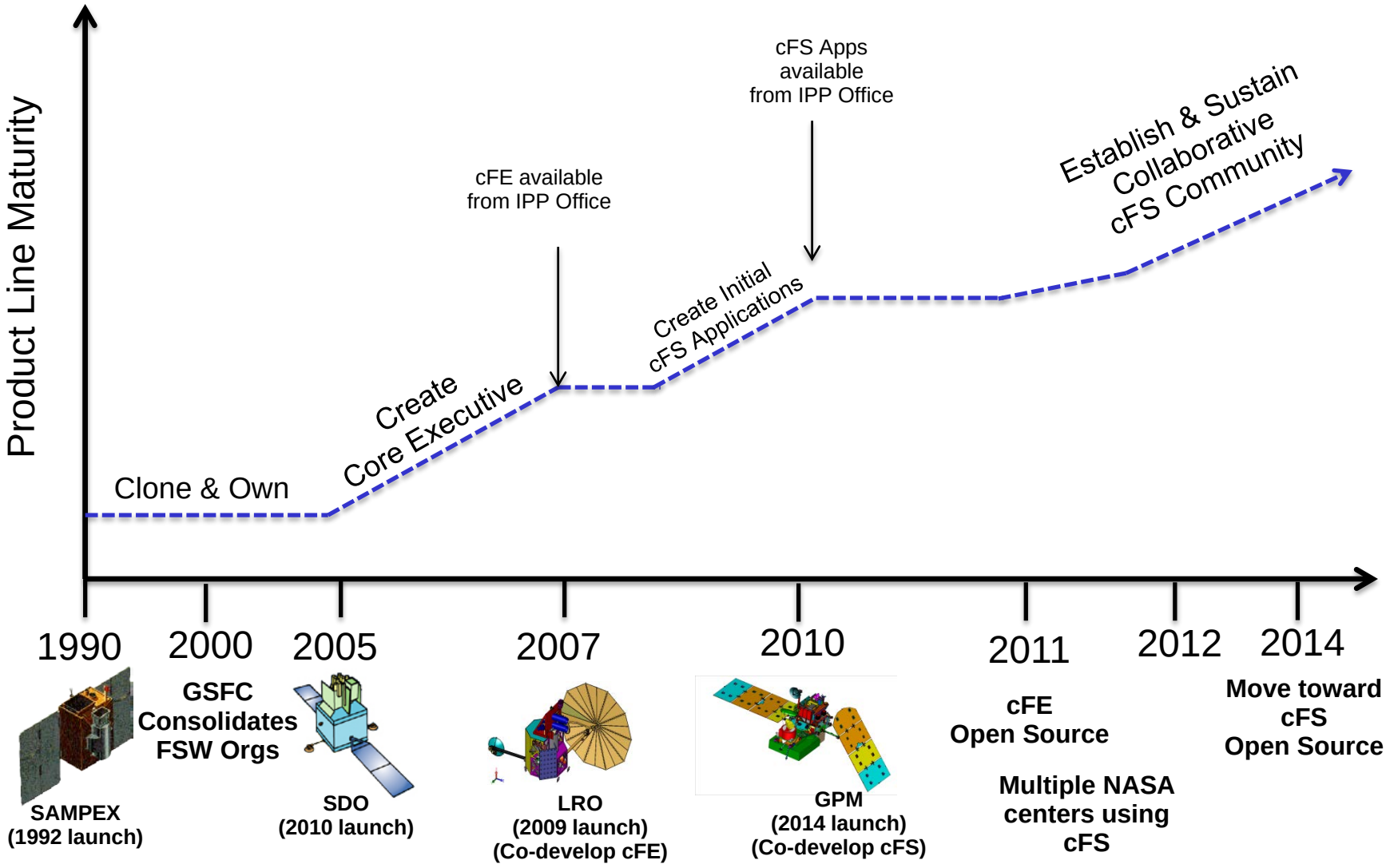
Realated Efforts

- NASA Operational Simulator for Small Satellites (NASA's Independent V&V Facility)
  - Ground System: Ball Aerospace's COSMOS
  - Dynamic Simulator: NASA Goddard's 42

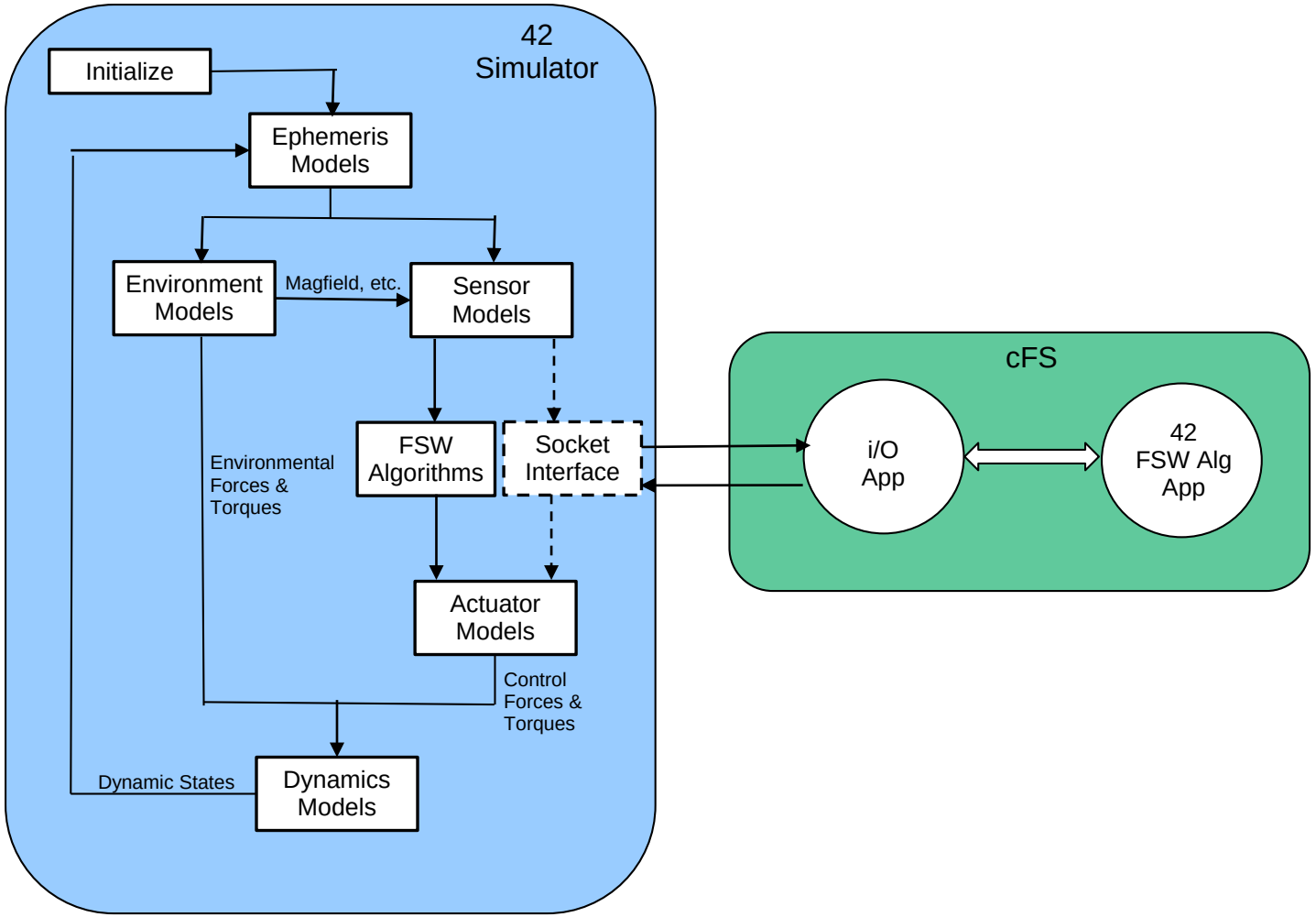
- The Hammers Company
  - Ground System: Hammer's Integrated Test and Operations System (ITOS)
  - Dynamic Simulator: Hammer's VIRTUALSAT®

- NASA Johnson Space Center
  - Ground System: Hammer's ITOS
  - Dynamic Simulator: NASA JSC Trick

cFS Timeline



42 Simulator Integration



Example CubeSat Application

Large Science Mission Flight Software

Vendor Software	Reuse Software	New Software
RTEMS OS	cFE Core	Bootstrap Loader
	cFS Scheduler	RTEMS Board Support Pkg.
	cFS Stored Commands	cFE Platform Support Pkg.
	cFS Health and Safety	Drivers / File Systems
	cFS Checksum	Command Ingest App
	cFS Limit Checker	Telemetry Output App
	cFS File Manager	Interface Manager App
	cFS Memory Dwell	Spacewire App/Driver
	cFS Memory Manager	Guidance / Nav App
	cFS Data Storage	Mission H/W Library
	cFS CFDP	Memory Scrub App
	cFS Housekeeping	Diagnostics Package
	ACS Math Library	

CubeSat Flight Software

Vendor Software	Reuse Software	New Software
FreeRTOS OS	cFE Core	FreeRTOS OSAL Layer
Bootstrap Loader	cFS Scheduler	cFE Platform Support Pkg.
Drivers	cFS Stored Commands	CADET Radio App
File Systems	cFS Health and Safety	DAGR Instrument App
	cFS Checksum	INMS Instrument App
	cFS Limit Checker	Reaction Wheel App
	cFS File Manager	ACS App
	cFS Memory Dwell	Ephemeris/GPS App
	cFS Memory Manager	Mission H/W Library
		File Uplink App File Downlink App
		Spacecraft Housekeeping App

The list seems long, but most of the apps are relatively simple